The Battle of the Neighborhoods

Mong Kok, Hong Kong

Chu Kai Ming

Introduction

This is part of the capstone project for the IBM Data Science professional program on Coursera. The report will analysis and visualize the data to find whether which venues should be recommended to the tourists according to their categories and popularity from public.

Mong Kok is a busy district located in Kowloon, Hong Kong. It is an area stacked with entertainment, food and beverages, local markets and big shopping malls. It is also well-known to be one of the best spots for tourists because of the neon light billboards. It was been described as the busiest district in the world by the Guinness World Records which has an extremely high density of population density of 130000/km2.

Business Problem

It is always great to start a business in Mong Kok as the population density is very high and traffic is very busy and you do not have to worry about no customers nearby. However, due to the extremely high rental payment in Hong Kong, especially in busy area like Mong Kok, it needs to be decided seriously and scientifically with analysis of data science.

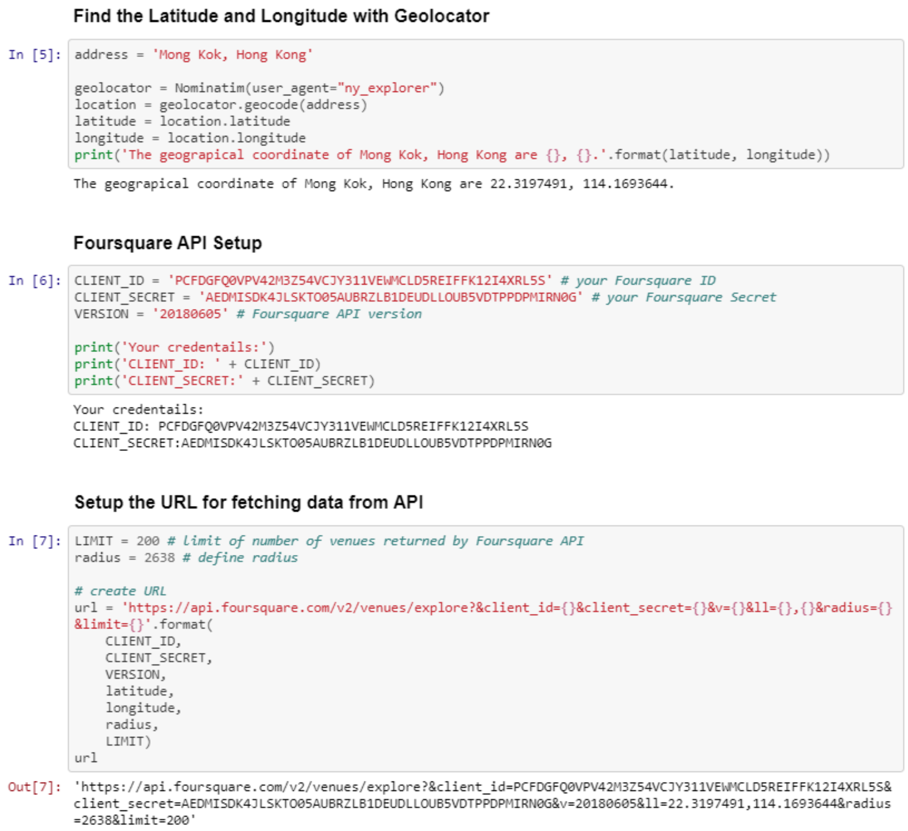
The target audience of this research would be stakeholders who would like to invest in the Mong Kok district, the stakeholders who would like to know more about this district with business insights and the people who need information to make business decisions.

Data

Data Collection

The data source of Mong Kok is mainly from the Foursquare API. The documentation of the API is in the following link: https://developer.foursquare.com/docs/api-reference/venues/search/

With these parts of code, we setup the link and credentials to fetch the data from Foursquare.



Find the Latitude and Longitude with Geolocator

```
In [5]: address = 'Mong Kok, Hong Kong'

        geolocator = Nominatim(user_agent="ny_explorer")
        location = geolocator.geocode(address)
        latitude = location.latitude
        longitude = location.longitude
        print('The geograpical coordinate of Mong Kok, Hong Kong are {}, {}.'.format(latitude, longitude))

        The geograpical coordinate of Mong Kok, Hong Kong are 22.3197491, 114.1693644.
```

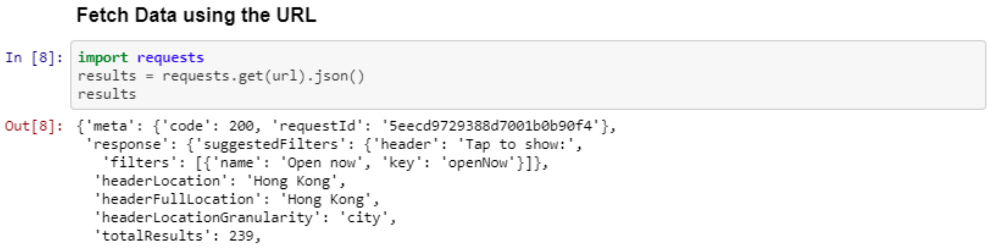
Foursquare API Setup

```
In [6]: CLIENT_ID = 'PCFDGFQ0VPV42M3Z54VCJY311VEWMCLD5REIFFK12I4XRL5S' # your Foursquare ID
        CLIENT_SECRET = 'AEDMISDK4JLSKTO05AUBRZLB1DEUDLLOUB5VDTPPDPMIRN0G' # your Foursquare Secret
        VERSION = '20180605' # Foursquare API version

        print('Your credentails:')
        print('CLIENT_ID: ' + CLIENT_ID)
        print('CLIENT_SECRET:' + CLIENT_SECRET)

        Your credentails:
        CLIENT_ID: PCFDGFQ0VPV42M3Z54VCJY311VEWMCLD5REIFFK12I4XRL5S
        CLIENT_SECRET:AEDMISDK4JLSKTO05AUBRZLB1DEUDLLOUB5VDTPPDPMIRN0G
```

Setup the URL for fetching data from API

```
In [7]: LIMIT = 200 # limit of number of venues returned by Foursquare API
        radius = 2638 # define radius

        # create URL
        url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&v={}&ll={},{}&radius={}
        &limit={}'.format(
            CLIENT_ID,
            CLIENT_SECRET,
            VERSION,
            latitude,
            longitude,
            radius,
            LIMIT)
        url

Out[7]: 'https://api.foursquare.com/v2/venues/explore?&client_id=PCFDGFQ0VPV42M3Z54VCJY311VEWMCLD5REIFFK12I4XRL5S&
        client_secret=AEDMISDK4JLSKTO05AUBRZLB1DEUDLLOUB5VDTPPDPMIRN0G&v=20180605&ll=22.3197491,114.1693644&radius
        =2638&limit=200'
```

Then, we fetch the data using the http requests to get the JSON format file.

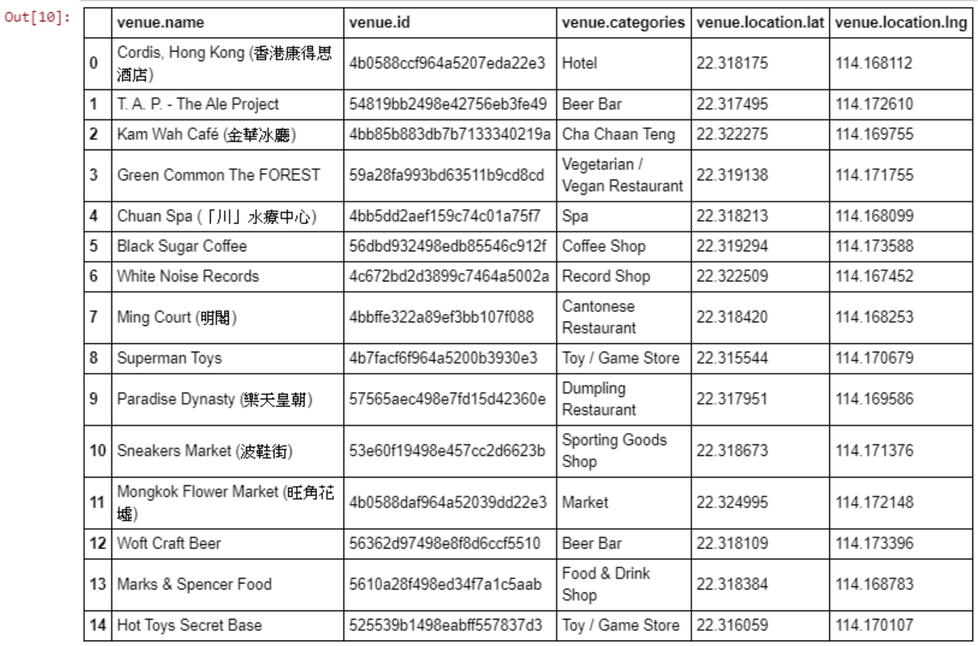


Fetch Data using the URL

```
In [8]: import requests
        results = requests.get(url).json()
        results

Out[8]: {'meta': {'code': 200, 'requestId': '5eecd9729388d7001b0b90f4'},
         'response': {'suggestedFilters': {'header': 'Tap to show:',
           'filters': [{'name': 'Open now', 'key': 'openNow'}]},
          'headerLocation': 'Hong Kong',
          'headerFullLocation': 'Hong Kong',
          'headerLocationGranularity': 'city',
          'totalResults': 239,
```

Data Pre-processing

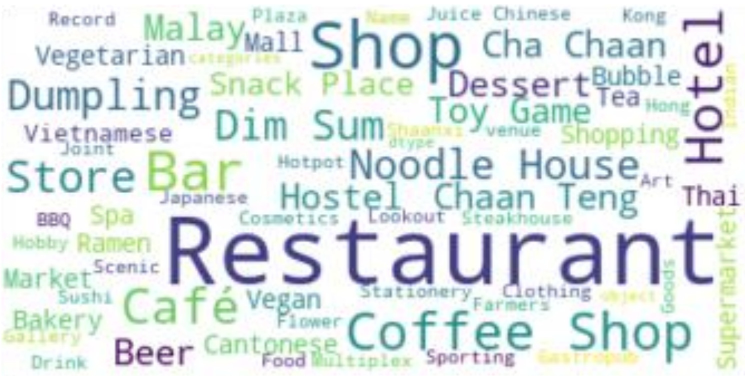
A long JSON file had been obtained through the HTTP request, but it is hard to be use without data pre-processing. Therefore, it is essential to turn the returned JSON file into a python dataframe before proceeding to the next step. The result dataframe contains the name of venues, a unique ID of the venues, the categories of the venues, and the latitude and longitude of the venues. The dataframe shows 100 venues and this graph is part of them.



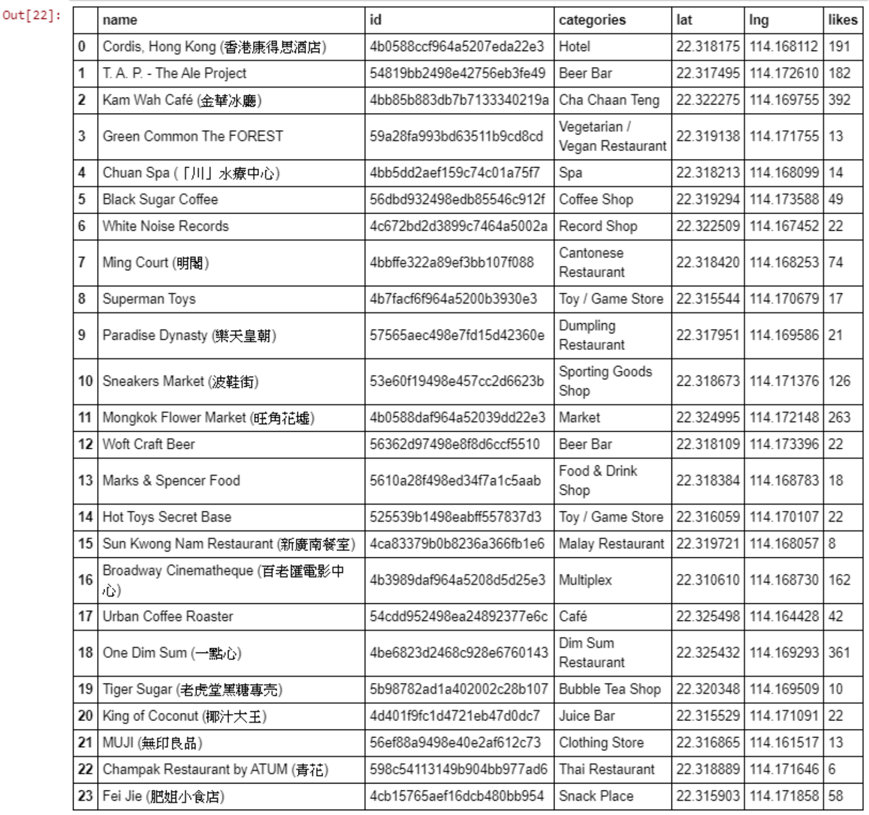
Out[10]:

	venue.name	venue.id	venue.categories	venue.location.lat	venue.location.lng
0	Cordis, Hong Kong (香港康得思酒店)	4b0588ccf964a5207eda22e3	Hotel	22.318175	114.168112
1	T. A. P. - The Ale Project	54819bb2498e42756eb3fe49	Beer Bar	22.317495	114.172610
2	Kam Wah Café (金華冰廳)	4bb85b883db7b7133340219a	Cha Chaan Teng	22.322275	114.169755
3	Green Common The FOREST	59a28fa993bd63511b9cd8cd	Vegetarian / Vegan Restaurant	22.319138	114.171755
4	Chuan Spa (「川」水療中心)	4bb5dd2aef159c74c01a75f7	Spa	22.318213	114.168099
5	Black Sugar Coffee	56dbd932498edb85546c912f	Coffee Shop	22.319294	114.173588
6	White Noise Records	4c672bd2d3899c7464a5002a	Record Shop	22.322509	114.167452
7	Ming Court (明閣)	4bbffe322a89ef3bb107f088	Cantonese Restaurant	22.318420	114.168253
8	Superman Toys	4b7facf6f964a5200b3930e3	Toy / Game Store	22.315544	114.170679
9	Paradise Dynasty (樂天皇朝)	57565aec498e7fd15d42360e	Dumpling Restaurant	22.317951	114.169586
10	Sneakers Market (波鞋街)	53e60f19498e457cc2d6623b	Sporting Goods Shop	22.318673	114.171376
11	Mongkok Flower Market (旺角花墟)	4b0588daf964a52039dd22e3	Market	22.324995	114.172148
12	Woft Craft Beer	56362d97498e8f8d6ccf5510	Beer Bar	22.318109	114.173396
13	Marks & Spencer Food	5610a28f498ed34f7a1c5aab	Food & Drink Shop	22.318384	114.168783
14	Hot Toys Secret Base	525539b1498eabff557837d3	Toy / Game Store	22.316059	114.170107

We found that there are 50 categories in total out of 100 venues. To have a better idea of what are these categories. I decided to generate a word cloud to have a better visualization on the data.



A list of number of likes get by these venues are also fetched from the Foursquare API. The list join as a new column into the dataframe. The column names are optimized to be shorter and easier to figure out. The new dataframe is look like this.

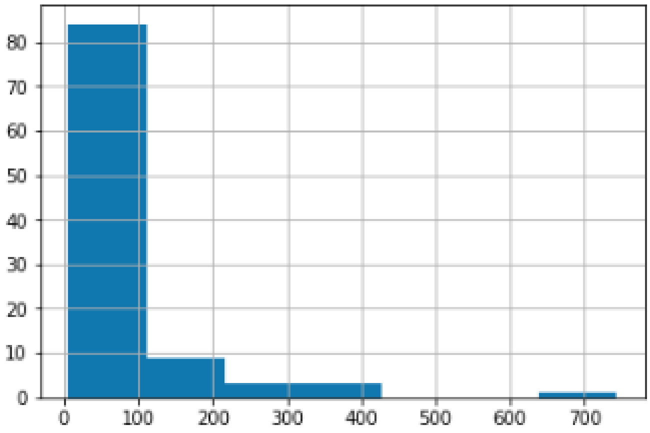


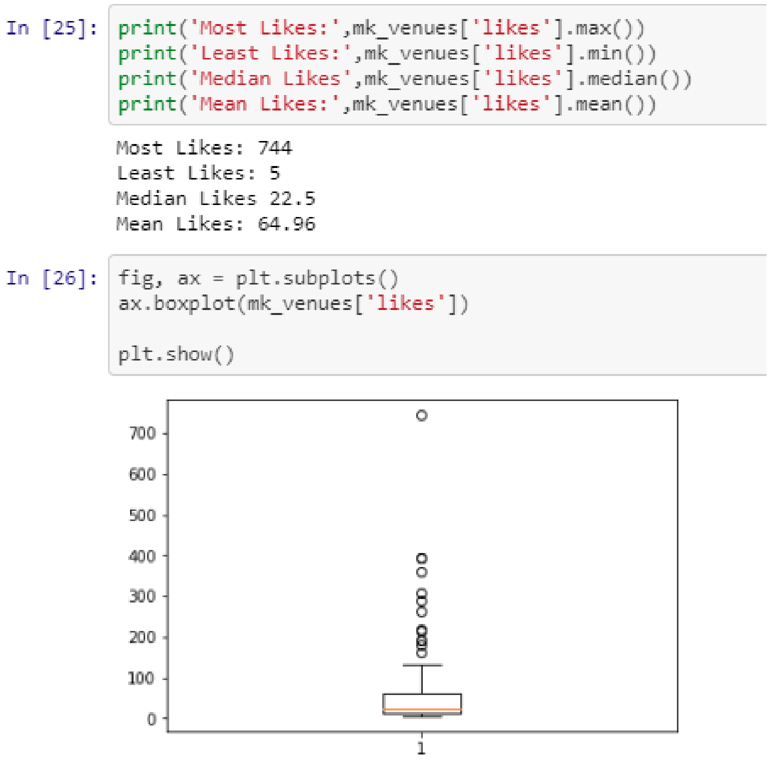
Out[22]:

	name	id	categories	lat	lng	likes
0	Cordis, Hong Kong (香港康得思酒店)	4b0588ccf964a5207eda22e3	Hotel	22.318175	114.168112	191
1	T. A. P. - The Ale Project	54819bb2498e42756eb3fe49	Beer Bar	22.317495	114.172610	182
2	Kam Wah Café (金華冰廳)	4bb85b883db7b7133340219a	Cha Chaan Teng	22.322275	114.169755	392
3	Green Common The FOREST	59a28fa993bd63511b9cd8cd	Vegetarian / Vegan Restaurant	22.319138	114.171755	13
4	Chuan Spa (「川」水療中心)	4bb5dd2aef159c74c01a75f7	Spa	22.318213	114.168099	14
5	Black Sugar Coffee	56dbd932498edb85546c912f	Coffee Shop	22.319294	114.173588	49
6	White Noise Records	4c672bd2d3899c7464a5002a	Record Shop	22.322509	114.167452	22
7	Ming Court (明閣)	4bbffe322a89ef3bb107f088	Cantonese Restaurant	22.318420	114.168253	74
8	Superman Toys	4b7facf6f964a5200b3930e3	Toy / Game Store	22.315544	114.170679	17
9	Paradise Dynasty (樂天皇朝)	57565aec498e7fd15d42360e	Dumpling Restaurant	22.317951	114.169586	21
10	Sneakers Market (波鞋街)	53e60f19498e457cc2d6623b	Sporting Goods Shop	22.318673	114.171376	126
11	Mongkok Flower Market (旺角花墟)	4b0588daf964a52039dd22e3	Market	22.324995	114.172148	263
12	Woft Craft Beer	56362d97498e8f8d6ccf5510	Beer Bar	22.318109	114.173396	22
13	Marks & Spencer Food	5610a28f498ed34f7a1c5aab	Food & Drink Shop	22.318384	114.168783	18
14	Hot Toys Secret Base	525539b1498eabff557837d3	Toy / Game Store	22.316059	114.170107	22
15	Sun Kwong Nam Restaurant (新廣南餐室)	4ca83379b0b8236a366fb1e6	Malay Restaurant	22.319721	114.168057	8
16	Broadway Cinematheque (百老匯電影中心)	4b3989daf964a5208d5d25e3	Multiplex	22.310610	114.168730	162
17	Urban Coffee Roaster	54cdd952498ea24892377e6c	Café	22.325498	114.164428	42
18	One Dim Sum (一點心)	4be6823d2468c928e6760143	Dim Sum Restaurant	22.325432	114.169293	361
19	Tiger Sugar (老虎堂黑糖專売)	5b98782ad1a402002c28b107	Bubble Tea Shop	22.320348	114.169509	10
20	King of Coconut (椰汁大王)	4d401f9fc1d4721eb47d0dc7	Juice Bar	22.315529	114.171091	22
21	MUJI (無印良品)	56ef88a9498e40e2af612c73	Clothing Store	22.316865	114.161517	13
22	Champak Restaurant by ATUM (青花)	598c54113149b904bb977ad6	Thai Restaurant	22.318889	114.171646	6
23	Fei Jie (肥姐小食店)	4cb15765aef16dcb480bb954	Snack Place	22.315903	114.171858	58

Data Visualization

In order to get better results from the clustering, further knowledges should be obtained before that. Histogram box diagram are generated by the likes that the number of venues get.



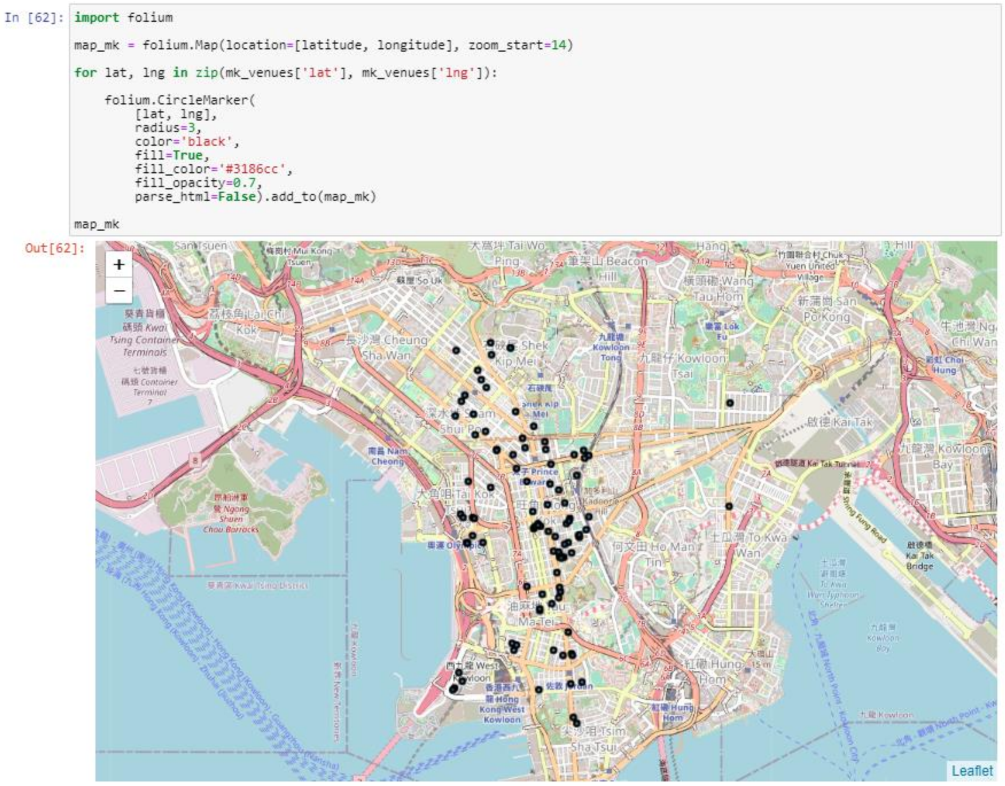


```
In [25]: print('Most Likes:',mk_venues['likes'].max())
         print('Least Likes:',mk_venues['likes'].min())
         print('Median Likes',mk_venues['likes'].median())
         print('Mean Likes:',mk_venues['likes'].mean())
```

```
Most Likes: 744
Least Likes: 5
Median Likes 22.5
Mean Likes: 64.96
```

```
In [26]: fig, ax = plt.subplots()
         ax.boxplot(mk_venues['likes'])

         plt.show()
```

Also, a map generated by folium visualize the location of the venues could give more information when making the data analysis decisions.

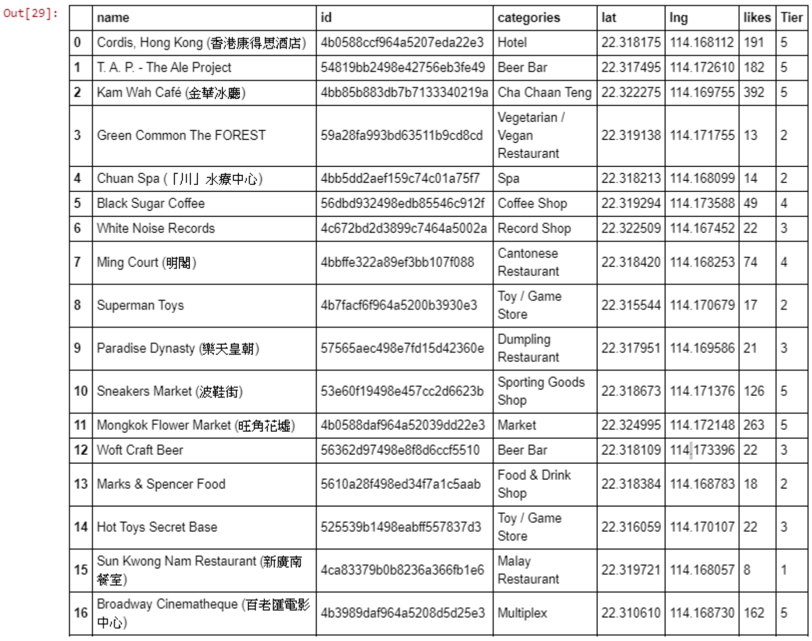


```
In [62]: import folium

         map_mk = folium.Map(location=[latitude, longitude], zoom_start=14)

         for lat, lng in zip(mk_venues['lat'], mk_venues['lng']):

             folium.CircleMarker(
                 [lat, lng],
                 radius=3,
                 color='black',
                 fill=True,
                 fill_color='#3186cc',
                 fill_opacity=0.7,
                 parse_html=False).add_to(map_mk)

         map_mk
```

Out[62]:

Data Labeling and Re-categorization

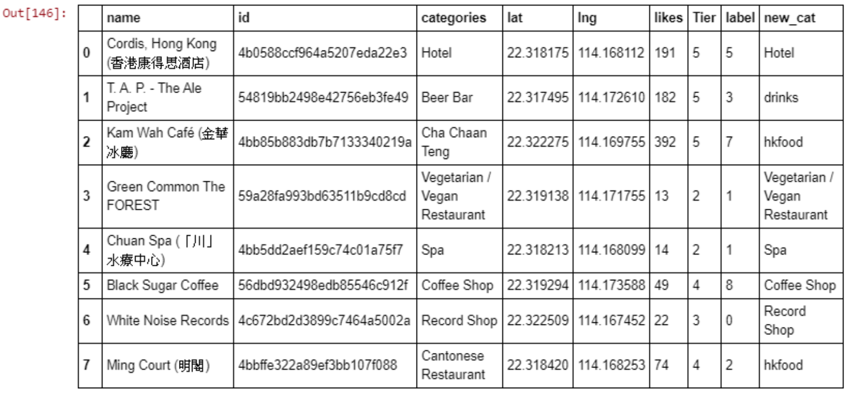
After visualizing the data from different ways, it is believed that labeling the venues with a 5-level tier list. The 5 tiers would be granted by the top 20%, 40%, 60%, 80% and 100% of the venues' like. A part of the new dataframe is in the following graph.



Out[29]:

	name	id	categories	lat	lng	likes	Tier
0	Cordis, Hong Kong (香港康得思酒店)	4b0588ccf964a5207eda22e3	Hotel	22.318175	114.168112	191	5
1	T. A. P. - The Ale Project	54819bb2498e42756eb3fe49	Beer Bar	22.317495	114.172610	182	5
2	Kam Wah Café (金華冰廳)	4bb85b883db7b7133340219a	Cha Chaan Teng	22.322275	114.169755	392	5
3	Green Common The FOREST	59a28fa993bd63511b9cd8cd	Vegetarian / Vegan Restaurant	22.319138	114.171755	13	2
4	Chuan Spa (「川」水療中心)	4bb5dd2aef159c74c01a75f7	Spa	22.318213	114.168099	14	2
5	Black Sugar Coffee	56dbd932498edb85546c912f	Coffee Shop	22.319294	114.173588	49	4
6	White Noise Records	4c672bd2d3899c7464a5002a	Record Shop	22.322509	114.167452	22	3
7	Ming Court (明閣)	4bbffe322a89ef3bb107f088	Cantonese Restaurant	22.318420	114.168253	74	4
8	Superman Toys	4b7facf6f964a5200b3930e3	Toy / Game Store	22.315544	114.170679	17	2
9	Paradise Dynasty (樂天皇朝)	57565aec498e7fd15d42360e	Dumpling Restaurant	22.317951	114.169586	21	3
10	Sneakers Market (波鞋街)	53e60f19498e457cc2d6623b	Sporting Goods Shop	22.318673	114.171376	126	5
11	Mongkok Flower Market (旺角花墟)	4b0588daf964a52039dd22e3	Market	22.324995	114.172148	263	5
12	Woft Craft Beer	56362d97498e8f8d6ccf5510	Beer Bar	22.318109	114.173396	22	3
13	Marks & Spencer Food	5610a28f498ed34f7a1c5aab	Food & Drink Shop	22.318384	114.168783	18	2
14	Hot Toys Secret Base	525539b1498eabff557837d3	Toy / Game Store	22.316059	114.170107	22	3
15	Sun Kwong Nam Restaurant (新廣南餐室)	4ca83379b0b8236a366fb1e6	Malay Restaurant	22.319721	114.168057	8	1
16	Broadway Cinematheque (百老匯電影中心)	4b3989daf964a5208d5d25e3	Multiplex	22.310610	114.168730	162	5

As there are too many categories (50) out of the total venues (100). Before proceeding to the next step, the categories require to be recategorize. There are many types of local restaurant selling local Hong Kong food but in different categories. Therefore, I recategorize these kind of restaurants into a new category called "hkfood". Also, there are many kinds of bars in the categories so they had been recategorize as "drinks". The new categories list is joined into the dataframe.



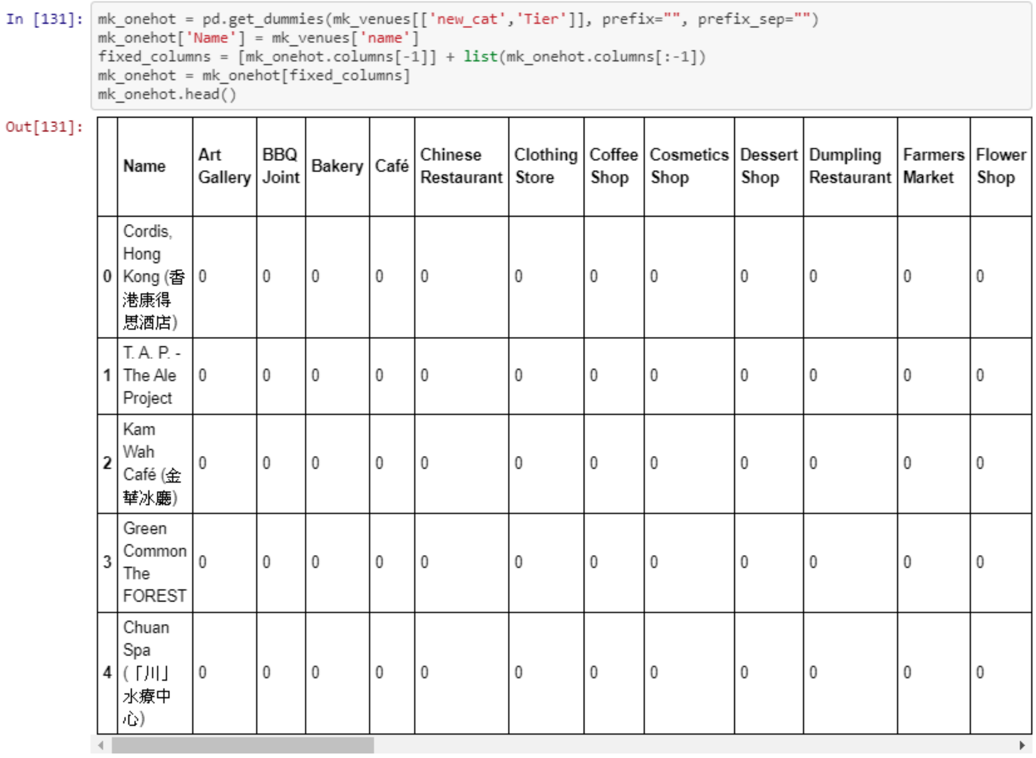
Out[146]:

	name	id	categories	lat	lng	likes	Tier	label	new_cat
0	Cordis, Hong Kong (香港康得思酒店)	4b0588ccf964a5207eda22e3	Hotel	22.318175	114.168112	191	5	5	Hotel
1	T. A. P. - The Ale Project	54819bb2498e42756eb3fe49	Beer Bar	22.317495	114.172610	182	5	3	drinks
2	Kam Wah Café (金華冰廳)	4bb85b883db7b7133340219a	Cha Chaan Teng	22.322275	114.169755	392	5	7	hkfood
3	Green Common The FOREST	59a28fa993bd63511b9cd8cd	Vegetarian / Vegan Restaurant	22.319138	114.171755	13	2	1	Vegetarian / Vegan Restaurant
4	Chuan Spa (「川」水療中心)	4bb5dd2aef159c74c01a75f7	Spa	22.318213	114.168099	14	2	1	Spa
5	Black Sugar Coffee	56dbd932498edb85546c912f	Coffee Shop	22.319294	114.173588	49	4	8	Coffee Shop
6	White Noise Records	4c672bd2d3899c7464a5002a	Record Shop	22.322509	114.167452	22	3	0	Record Shop
7	Ming Court (明閣)	4bbffe322a89ef3bb107f088	Cantonese Restaurant	22.318420	114.168253	74	4	2	hkfood

Methodology

One Hot Encoding

The new categories and the tier list would be proceed into the one hot encoding and be the main attributes of the

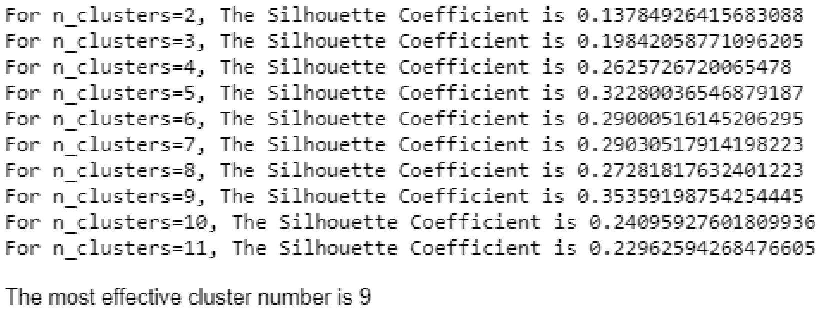


```
In [131]: mk_onehot = pd.get_dummies(mk_venues[['new_cat','Tier']], prefix="", prefix_sep="")
          mk_onehot['Name'] = mk_venues['name']
          fixed_columns = [mk_onehot.columns[-1]] + list(mk_onehot.columns[:-1])
          mk_onehot = mk_onehot[fixed_columns]
          mk_onehot.head()
```

Out[131]:

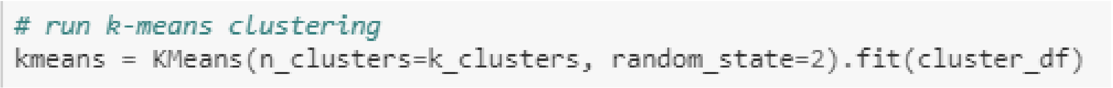
	Name	Art Gallery	BBQ Joint	Bakery	Café	Chinese Restaurant	Clothing Store	Coffee Shop	Cosmetics Shop	Dessert Shop	Dumpling Restaurant	Farmers Market	Flower Shop
0	Cordis, Hong Kong (香港康得思酒店)	0	0	0	0	0	0	0	0	0	0	0	0
1	T. A. P. - The Ale Project	0	0	0	0	0	0	0	0	0	0	0	0
2	Kam Wah Café (金華冰廳)	0	0	0	0	0	0	0	0	0	0	0	0
3	Green Common The FOREST	0	0	0	0	0	0	0	0	0	0	0	0
4	Chuan Spa (「川」水療中心)	0	0	0	0	0	0	0	0	0	0	0	0

Find the most effective cluster number with The Silhouette Coefficient



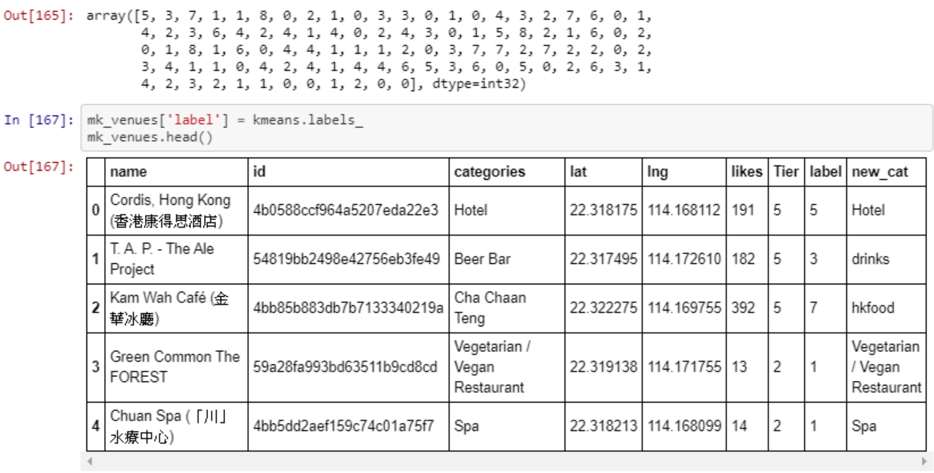
```
For n_clusters=2, The Silhouette Coefficient is 0.13784926415683088
For n_clusters=3, The Silhouette Coefficient is 0.19842058771096205
For n_clusters=4, The Silhouette Coefficient is 0.2625726720065478
For n_clusters=5, The Silhouette Coefficient is 0.32280036546879187
For n_clusters=6, The Silhouette Coefficient is 0.29000516145206295
For n_clusters=7, The Silhouette Coefficient is 0.29030517914198223
For n_clusters=8, The Silhouette Coefficient is 0.27281817632401223
For n_clusters=9, The Silhouette Coefficient is 0.35359198754254445
For n_clusters=10, The Silhouette Coefficient is 0.24095927601809936
For n_clusters=11, The Silhouette Coefficient is 0.22962594268476605

The most effective cluster number is 9
```

The k-clusters is set to 9 and the k means labels are generated.



```
# run k-means clustering
kmeans = KMeans(n_clusters=k_clusters, random_state=2).fit(cluster_df)
```

Results



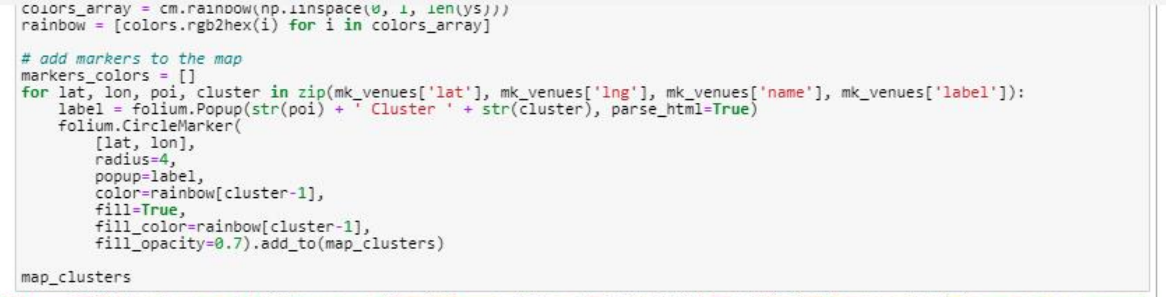
```
Out[165]: array([5, 3, 7, 1, 1, 8, 0, 2, 1, 0, 3, 3, 0, 1, 0, 4, 3, 2, 7, 6, 0, 1,
                 4, 2, 3, 6, 4, 2, 4, 1, 4, 0, 2, 4, 3, 0, 1, 5, 8, 2, 1, 6, 0, 2,
                 0, 1, 8, 1, 6, 0, 4, 4, 1, 1, 1, 2, 0, 3, 7, 7, 2, 7, 2, 2, 0, 2,
                 3, 4, 1, 1, 0, 4, 2, 4, 1, 4, 4, 6, 5, 3, 6, 0, 5, 0, 2, 6, 3, 1,
                 4, 2, 3, 2, 1, 1, 0, 0, 1, 2, 0, 0], dtype=int32)
```

```
In [167]: mk_venues['label'] = kmeans.labels_
          mk_venues.head()
```

Out[167]:

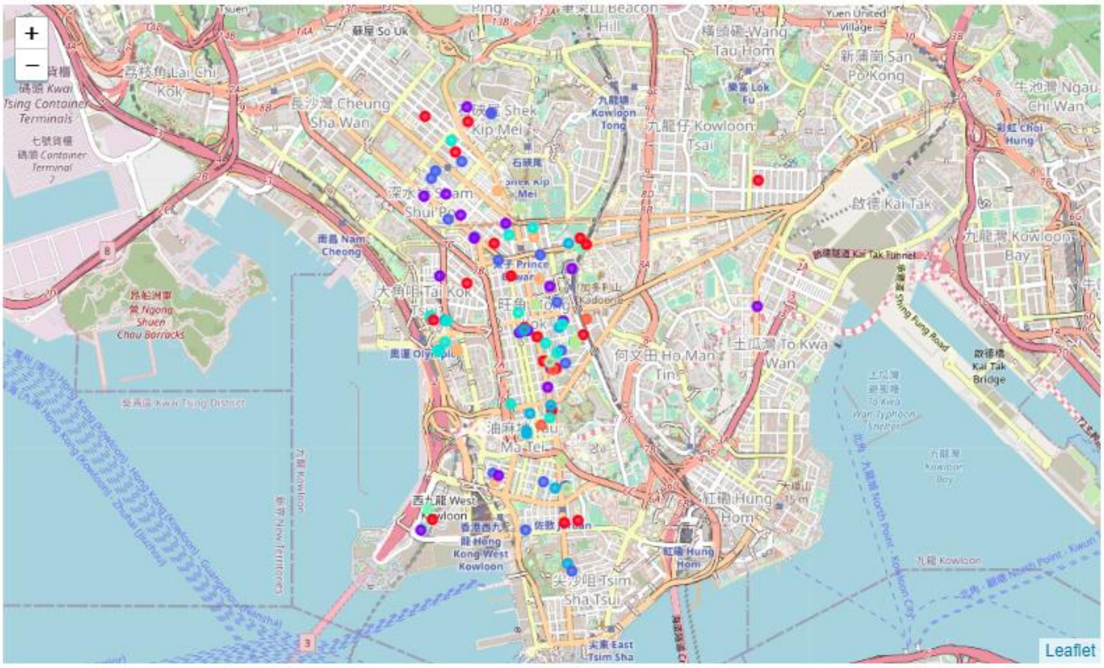
	name	id	categories	lat	lng	likes	Tier	label	new_cat
0	Cordis, Hong Kong (香港康得思酒店)	4b0588ccf964a5207eda22e3	Hotel	22.318175	114.168112	191	5	5	Hotel
1	T. A. P. - The Ale Project	54819bb2498e42756eb3fe49	Beer Bar	22.317495	114.172610	182	5	3	drinks
2	Kam Wah Café (金華冰廳)	4bb85b883db7b7133340219a	Cha Chaan Teng	22.322275	114.169755	392	5	7	hkfood
3	Green Common The FOREST	59a28fa993bd63511b9cd8cd	Vegetarian / Vegan Restaurant	22.319138	114.171755	13	2	1	Vegetarian / Vegan Restaurant
4	Chuan Spa (「川」水療中心)	4bb5dd2aef159c74c01a75f7	Spa	22.318213	114.168099	14	2	1	Spa

The above dataframe is the results of the clustering and the following map is generated by folium with different colors according to the labels they get from different clustering groups.



```
colors_array = cm.rainbow(np.linspace(0, 1, len(ys)))
rainbow = [colors.rgb2hex(i) for i in colors_array]

# add markers to the map
markers_colors = []
for lat, lon, poi, cluster in zip(mk_venues['lat'], mk_venues['lng'], mk_venues['name'], mk_venues['label']):
    label = folium.Popup(str(poi) + ' Cluster ' + str(cluster), parse_html=True)
    folium.CircleMarker(
        [lat, lon],
        radius=4,
        popup=label,
        color=rainbow[cluster-1],
        fill=True,
        fill_color=rainbow[cluster-1],
        fill_opacity=0.7).add_to(map_clusters)

map_clusters
```

68]:



The results of cluster groups are long lists and would not be shown in this report. Check out the jupyter notebook on GitHub to see the complete results and data analysis processes:

https://github.com/siuyuk/IBM-Data-Sci-Capstone-Project/blob/master/mongkokseg%26cluster.ipynb

Conclusions

The data source of this project is not big and complete enough to make it to be great. I tried my best to use these resources on hands to get the above results with the knowledges I learnt from the 9 courses. The target audience of this report should get more information about the venues that should be recommended in different clusters. If I could have some improvements for the analysis, I would try to find more data or change the attributes I used for one hot encoding and clustering number to finalize the results better.